## 1.Introduction to SQL and installation of SQL Server / Oracle.

## What is DataBase:

## Database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS)Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database.

## Data within the most common types of databases in operation today is typically modeled in rows and columns in a series of tables to make processing and data querying efficient. The data can then be easily accessed, managed, modified, updated, controlled, and organized. Most databases use structured query language (SQL) for writing and querying data.

## Types of databases

## There are many different types of databases. The best database for a specific organization depends on how the organization intends to use the data.

## Relational databases

## Relational Databases became dominant in the 1980s. Items in a relational database are organized as a set of tables with columns and rows. Relational database technology provides the most efficient and flexible way to access structured information.

## Object-oriented databases

## Information in an object-oriented database is represented in the form of objects, as in object-oriented programming.

## Distributed databases

## A distributed database consists of two or more files located in different sites. The database may be stored on multiple computers, located in the same physical location, or scattered over different networks.

## Data warehouses

## A central repository for data, a data warehouse is a type of database specifically designed for fast query and analysis.

## NoSQL databases

## A NoSQL, or nonrelational database, allows unstructured and semistructured data to be stored and manipulated (in contrast to a relational database, which defines how all data inserted into the database must be composed). NoSQL databases grew popular as web applications became more common and more complex.

## Graph databases

## A graph database stores data in terms of entities and the relationships between entities.

## ****OLTP databases**.** An OLTP database is a speedy, analytic database designed for large numbers of transactions performed by multiple users.

## These are only a few of the several dozen types of databases in use today. Other, less common databases are tailored to very specific scientific, financial, or other functions. In addition to the different database types, changes in technology development approaches and dramatic advances such as the cloud and automation are propelling databases in entirely new directions. Some of the latest databases include :

## Open source databases

## An open source database system is one whose source code is open source; such databases could be SQL or NoSQL databases.

## Cloud databases

## A cloud database is a collection of data, either structured or unstructured, that resides on a private, public, or hybrid cloud computing platform. There are two types of cloud database models: traditional and database as a service (DBaaS). With DBaaS, administrative tasks and maintenance are performed by a service provider.

## Multimodel database

## Multimodel databases combine different types of database models into a single, integrated back end. This means they can accommodate various data types.

## Document/JSON database

## Designed for storing, retrieving, and managing document-oriented information, document databses are a modern way to store data in JSON format rather than rows and columns.

## Self-driving databases

## The newest and most groundbreaking type of database, self-driving databases (also known as autonomous databases) are cloud-based and use machine learning to automate database tuning, security, backups, updates, and other routine management tasks traditionally performed by database administrators.

## What is Structured Query Language (SQL)?

## SQL is a programming language used by nearly all relational databases to query, manipulate, and define data, and to provide access control. SQL was first developed at IBM in the 1970s with Oracle as a major contributor, which led to implementation of the SQL ANSI standard, SQL has spurred many extensions from companies such as IBM, Oracle, and Microsoft. Although SQL is still widely used today, new programming languages are beginning to appear.

## Applications of SQL

## As mentioned before, SQL is one of the most widely used query language over the databases. I'm going to list few of them here:

## Allows users to access data in the relational database management systems.

## Allows users to describe the data.

## Allows users to define the data in a database and manipulate that data.

## Allows to embed within other languages using SQL modules, libraries & pre-compilers.

## Allows users to create and drop databases and tables.

## Allows users to create view, stored procedure, functions in a database.

## Allows users to set permissions on tables, procedures and views.

## What is a MySQL database?

## MySQL is an open source relational database management system based on SQL. It was designed and optimized for web applications and can run on any platform. As new and different requirements emerged with the internet, MySQL became the platform of choice for web developers and web-based applications. Because it’s designed to process millions of queries and thousands of transactions, MySQL is a popular choice for ecommerce businesses that need to manage multiple money transfers. On-demand flexibility is the primary feature of MySQL.

## MySQL is the DBMS behind some of the top websites and web-based applications in the world, including Airbnb, Uber, LinkedIn, Facebook, Twitter, and YouTube.

## Features:

## High Performance.

## High Availability.

## Scalability and Flexibility Run anything.

## Robust Transactional Support.

## Web and Data Warehouse Strengths.

## Strong Data Protection.

## Comprehensive Application Development.

## Management Ease.

## Open Source Freedom and 24 x 7 Support.

## Lowest Total Cost of Ownership.

## Features Added in MySQL 8.0

## The following features are added in MySQL 8.0 version:

## ****Data Dictionary:**** It incorporates the transactional data dictionary to stores information about the database objects. Previous versions stored data in metadata files and non-transactional tables.

## ****Atomic DDL Statement:**** It is an Atomic Data Definition Language statement that combines storage engine operations, data dictionary updates, and binary log associated with a DDL operation into a single atomic transaction.

## ****Upgrade Procedures:**** Previously, the installation of the new MySQL version automatically upgrade the data dictionary table at the next startup, and then DBA is expected to invoke mysql\_upgrade command manually for completing the upgrading process. After MySQL 8.0.16, it is not dependent on the DBA to invoke mysql\_upgrade command for completing the up-gradation process.

## ****Security and account management:**** There is some enhancement added to improve the security and provide it to enable greater DBA flexibility in account management.

## ****Resource Management:**** Now, MySQL allows you to create and support resource groups, assign threads to a particular group so that it can execute according to the resource available for the group. Group attributes can control its resource consumption by threads in the group.

## ****Table Encryption Management:**** Now, table encryption is managed globally by defining and enforcing encryption defaults. The default\_table\_encryption variable or DEFAULT ENCRYPTION clause defines encryption default when creating a schema and general tablespace.

## ****InnoDB enhancements:**** The InnoDB enhancement were added in auto-increment counter, index tree corruption, memcached plugin, InnoDB\_deadlock\_detect, tablespace encryption feature, storage engine, InnoDB\_dedicated\_server, zlib library, and many more.

## ****Character Set Support:**** The default character set now changed from latin1 to utf8mb4. The new character set has many new collations, including utf8mb\_ja\_0900\_as\_cs.

## ****JSON Enhancements:**** The following enhancements or additions are introduced in the MySQL’s json functionality: Inline path (->>) operator, json aggregate functions JSON\_ARRAYAGG() and JSON\_OBJECTAGG(), utility function JSON\_PRETTY(), JSON\_STORAGE\_SIZE()(, JSON\_STORAGE\_FREE(). In sorting json values, now each value is represented by variable-length part of sort key instead of a fixed 1K size. It also added merge function JSON\_MERGE\_PATCH to add 2 json object and JSON\_TABLE() function.

## ****Data Type Support:**** In data type specifications, it can support the use of expressions as default values.

## ****Optimizer Enhancement:**** This version added optimizer enhancement such as invisible indexes, descending indexes, support the creation of a functional index. It can use constant folding for comparison between a column and a constant value.

## ****Window Function:**** This version supports many new window functions such as RANK(), LAG(), and NTILE().

## Some other important features are:

## It enhances Regular expression support.

## Error Logging re-written to use MySQL component architecture.

## A new backup lock introduced that permits DML while preventing an operation, which can result in an inconsistent state.

## It enhances connection management. Now, TCP/IP port can be configured specifically for administrative connections. It gives more control in compression to minimize the bytes sent over the connection to the server.

## In previous versions, the plugins were written in C or C++. Now, it must be written in only the C++ language. MySQL 8.0.17 version provides clone plugins, which permit InnoDB data locally or from a remote server. The clone plugin also supports replication.

## In this version, the time zone support for TIMESTAMP and DATETIME values.

## This version also added the SQL standard table value constructor and explicit table clause.

## Use of mysql:

## ****1. Scalability and Flexibility****

## The MySQL database server provides the ultimate in scalability, sporting the capacity to handle deeply embedded applications with a footprint of only 1MB to running massive data warehouses holding terabytes of information. Platform flexibility is a stalwart feature of MySQL with all flavors of Linux, UNIX, and Windows being supported. And, of course, the open source nature of MySQL allows complete customization for those wanting to add unique requirements to the database server.

## ****2. High Performance****

## A unique storage-engine architecture allows database professionals to configure the MySQL database server specifically for particular applications, with the end result being amazing performance results. Whether the intended application is a high-speed transactional processing system or a high-volume web site that services a billion queries a day, MySQL can meet the most demanding performance expectations of any system. With high-speed load utilities, distinctive memory caches, full text indexes, and other performance-enhancing mechanisms, MySQL offers all the right ammunition for today's critical business systems.

## ****3. High Availability****

## Rock-solid reliability and constant availability are hallmarks of MySQL, with customers relying on MySQL to guarantee around-the-clock uptime. MySQL offers a variety of high-availability options from high-speed master/slave replication configurations, to specialized Cluster servers offering instant failover, to third party vendors offering unique high-availability solutions for the MySQL database server.

## ****4. Robust Transactional Support****

## MySQL offers one of the most powerful transactional database engines on the market. Features include complete ACID (atomic, consistent, isolated, durable) transaction support, unlimited row-level locking, distributed transaction capability, and multi-version transaction support where readers never block writers and vice-versa. Full data integrity is also assured through server-enforced referential integrity, specialized transaction isolation levels, and instant deadlock detection.

## ****5. Web and Data Warehouse Strengths****

## MySQL is the de-facto standard for high-traffic web sites because of its high-performance query engine, tremendously fast data insert capability, and strong support for specialized web functions like fast full text searches. These same strengths also apply to data warehousing environments where MySQL scales up into the terabyte range for either single servers or scale-out architectures. Other features like main memory tables, B-tree and hash indexes, and compressed archive tables that reduce storage requirements by up to eighty-percent make MySQL a strong standout for both web and business intelligence applications.

## ****6. Strong Data Protection****

## Because guarding the data assets of corporations is the number one job of database professionals, MySQL offers exceptional security features that ensure absolute data protection. In terms of database authentication, MySQL provides powerful mechanisms for ensuring only authorized users have entry to the database server, with the ability to block users down to the client machine level being possible. SSH and SSL support are also provided to ensure safe and secure connections. A granular object privilege framework is present so that users only see the data they should, and powerful data encryption and decryption functions ensure that sensitive data is protected from unauthorized viewing. Finally, backup and recovery utilities provided through MySQL and third party software vendors allow for complete logical and physical backup as well as full and point-in-time recovery.

## ****7. Comprehensive Application Development****

## One of the reasons MySQL is the world's most popular open source database is that it provides comprehensive support for every application development need. Within the database, support can be found for stored procedures, triggers, functions, views, cursors, ANSI-standard SQL, and more. For embedded applications, plug-in libraries are available to embed MySQL database support into nearly any application. MySQL also provides connectors and drivers (ODBC, JDBC, etc.) that allow all forms of applications to make use of MySQL as a preferred data management server. It doesn't matter if it's PHP, Perl, Java, Visual Basic, or .NET, MySQL offers application developers everything they need to be successful in building database-driven information systems.

## ****8. Management Ease****

## MySQL offers exceptional quick-start capability with the average time from software download to installation completion being less than fifteen minutes. This rule holds true whether the platform is Microsoft Windows, Linux, Macintosh, or UNIX. Once installed, self-management features like automatic space expansion, auto-restart, and dynamic configuration changes take much of the burden off already overworked database administrators. MySQL also provides a complete suite of graphical management and migration tools that allow a DBA to manage, troubleshoot, and control the operation of many MySQL servers from a single workstation. Many third party software vendor tools are also available for MySQL that handle tasks ranging from data design and ETL, to complete database administration, job management, and performance monitoring.

## ****9. Open Source Freedom and 24 x 7 Support****

## Many corporations are hesitant to fully commit to open source software because they believe they can't get the type of support or professional service safety nets they currently rely on with proprietary software to ensure the overall success of their key applications. The questions of indemnification come up often as well. These worries can be put to rest with MySQL as complete around-the-clock support as well as indemnification is available through MySQL Enterprise. MySQL is not a typical open source project as all the software is owned and supported by Oracle, and because of this, a unique cost and support model are available that provides a unique combination of open source freedom and trusted software with support.

## ****10. Lowest Total Cost of Ownership****

## By migrating current database-drive applications to MySQL, or using MySQL for new development projects, corporations are realizing cost savings that many times stretch into seven figures. Accomplished through the use of the MySQL database server and scale-out architectures that utilize low-cost commodity hardware, corporations are finding that they can achieve amazing levels of scalability and performance, all at a cost that is far less than those offered by proprietary and scale-up software vendors. In addition, the reliability and easy maintainability of MySQL means that database administrators don't waste time troubleshooting performance or downtime issues, but instead can concentrate on making a positive impact on higher level tasks that involve the business side of data.

## Sql commands:

## Some of The Most Important SQL Commands

## SELECT - extracts data from a database

## UPDATE - updates data in a database

## DELETE - deletes data from a database

## INSERT INTO - inserts new data into a database

## CREATE DATABASE - creates a new database

## ALTER DATABASE - modifies a database

## CREATE TABLE - creates a new table

## ALTER TABLE - modifies a table

## DROP TABLE - deletes a table

## CREATE INDEX - creates an index (search key)

## DROP INDEX - deletes an index

## The SQL SELECT Statement

## The SELECT statement is used to select data from a database.

## The data returned is stored in a result table, called the result-set.

## SELECT Syntax:

## SELECT column1*,* column2, ...

## FROM table\_name*;*

## Here, column1, column2, ... are the field names of the table you want to select data from. If you want to select all the fields available in the table, use the following syntax:

## SELECT\*FROM table\_name;

## The SQL UPDATE Statement

## The UPDATE statement is used to modify the existing records in a table.

## UPDATE Syntax:

## UPDATE table\_name

## SET column1 *=* value1*,* column2 *=* value2*, ...*

## WHERE condition;

## The SQL DELETE Statement

## The DELETE statement is used to delete existing records in a table.

## DELETE Syntax

## DELETE FROM table\_name WHERE condition;

## The SQL INSERT INTO Statement

## The INSERT INTO statement is used to insert new records in a table.

## INSERT INTO Syntax

## It is possible to write the INSERT INTO statement in two ways:

## 1. Specify both the column names and the values to be inserted:

## INSERT INTO table\_name(column1*,* column2*,* column3*,*…)

## VALUES (value1*,* value2*,* value3, ...);

## 2. If you are adding values for all the columns of the table, you do not need to specify the column names in the SQL query. However, make sure the order of the values is in the same order as the columns in the table. Here, the INSERT INTO syntax would be as follows:

## INSERT INTO table\_name

## VALUES (value1*,* value2*,* value3, ...);

## The SQL CREATE DATABASE Statement

## The CREATE DATABASE statement is used to create a new SQL database.

## Syntax

## CREATE DATABASE databasename;

## The SQL CREATE TABLE Statement

## The CREATE TABLE statement is used to create a new table in a database.

## Syntax

## CREATE TABLE table\_name (     column1 datatype,     column2 datatype,     column3 datatype,    .... );

## SQL ALTER TABLE Statement

## The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

## The ALTER TABLE statement is also used to add and drop various constraints on an existing table.

## ALTER TABLE - ADD Column

## To add a column in a table, use the following syntax:

## ALTER TABLE table\_name ADD column\_name datatype;

## The SQL DROP TABLE Statement

## The DROP TABLE statement is used to drop an existing table in a database.

## Syntax

## DROP TABLE table\_name;

## SQL CREATE INDEX Statement

## The CREATE INDEX statement is used to create indexes in tables.

## Indexes are used to retrieve data from the database more quickly than otherwise. The users cannot see the indexes, they are just used to speed up searches/queries.

## CREATE INDEX Syntax

## Creates an index on a table. Duplicate values are allowed:

## CREATE INDEX index\_name

## ON table\_name (column1*,* column2, ...);

## DROP INDEX Statement

## The DROP INDEX statement is used to delete an index in a table.

## MS Access:

## DROP INDEX index\_name ON table\_name;

## SQL Server:

## DROP INDEX table\_name.index\_name;

## DB2/Oracle:

## DROP INDEX index\_name;

## MySQL:

## ALTER TABLE table\_name DROP INDEX index\_name;